Simplified swarm optimization for feasibility-based rules on constrained engineering design optimization

Wei-Chang Yeh\textsuperscript{1} and Chia-Ling Huang\textsuperscript{2}

\textsuperscript{1}Distinguished Professor, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan  
yeh@ieee.org

\textsuperscript{2}Associate Professor, Department of Logistics and Shipping Management, Kainan University, Taoyuan, Taiwan  
Corresponding author’s E-mail: clhuang@mail.knu.edu.tw

Abstract: The constrained optimization benchmarks can successfully formulate various practical engineering designs. Therefore, there are various algorithms have been applied to optimize the engineering design by the formulation of the constrained optimization benchmarks since the last decades. The value of the solutions of any objective function that is obtained by any algorithm is ensured only if the solutions of the constraints are falling in feasible solutions. Therefore, it is very important to make sure the solutions are feasible that is the constraints are satisfied before declaring the objective solution obtained by any algorithm. In this work, we use a population-based swarm intelligence algorithm, called simplified swarm optimization (SSO) to optimize the constrained engineering design benchmarks based on the feasible solutions. For the purpose to evaluate the optimization performance, the SSO has performed on three well known constrained engineering design benchmarks including two different types of minimization constrained benchmarks and one engineering constrained and mechanical design benchmark. The computational results have been compared favourably with those obtained using existing algorithms in the literatures. The comparison results demonstrate the proposed SSO well optimize the benchmarks with feasible solutions compared to the other considered algorithms.

Keywords: simplified swarm optimization (SSO), constrained engineering design, optimization

1. INTRODUCTION

The constrained optimization benchmarks can successfully formulate various practical engineering designs. Therefore, there are various algorithms, such as mine blast algorithm (MBA), harmony search (HS), genetic algorithm (GA), evolutionary programming (EP), improved genetic algorithm (IGA), cultured differential evolution (CA-DE), co-evolutionary particle swarm optimization (CPSO), hybrid particle swarm optimization (HPSO), Nelder-Mead simplex search and particle swarm optimization (NM-PSO), quantum-behaved particle swarm optimization (Q-PSO), and Combining multiobjective optimization with differential evolution (C-DE), have been applied to optimize the engineering design by the formulation of the constrained optimization benchmarks since the last decades [1-15].

As we know, the optimization benchmarks in the real-world practice must be with the constraints. That is all of the optimization engineering design benchmarks are subjected under various
Simplified swarm optimization for feasibility-based rules on constrained engineering design optimization

Yeh

constraints. One of the most important techniques is to handle constraints because optimization value obtained by any algorithm seriously losses if the solutions of constraints fall into infeasible regions. Therefore, constraint handling is one of the major concerns when applying the algorithms to solve constrained engineering design optimization benchmarks. First priority, this paper proposes to exam the constraints for feasible solutions or infeasible solutions before comparison the objective solution obtained among each algorithm. Then, the real value of the objective solution obtained by any algorithm can be ensured if having the feasible solutions of constraints.

In this paper, the simplified swarm optimization (SSO) algorithm that belongs to the population-based swarm intelligence is proposed to optimize constrained engineering design benchmarks based on feasible solutions. SSO algorithm is developed and first introduced by Yeh [16] in 2009. SSO has been successfully exploited in considerable studies and applied in various fields. Among these, numerical varieties of constrained engineering design problems have been optimized by SSO, such as optimization of the Disassembly Sequencing Problems [17-19], the series-parallel redundancy allocation problems [20], forecasting wind power [21], and reliability redundancy allocation problems [22].

To demonstrate the performance of SSO, three benchmarks including two different types of minimization constrained benchmarks and one engineering constrained and mechanical design benchmark are considered and demonstrated in this study. The constrained solutions are examined for feasible or infeasible first to ensure the real value of the objective solutions obtained by each algorithm. The experimental solutions are compared with those of previously developed algorithms in literature, the results indicate the proposed SSO performs well for handling feasible solutions of constraints.

2. SIMPLIFIED SWARM OPTIMIZATION (SSO)

SSO is a novel developed soft computing algorithm and first introduced by Yeh [16]. The SSO makes use of a new update mechanism, shown in Eq. (1). The parameter $C_u=C_u$, $C_p=C_u+C_p$, and $C_g=C_g+C_g$, where $C_u$, $C_p$, and $C_g$ are the values of probability; $\rho \in \text{Uniform}(0,1)$ and $x \in \text{Uniform}(l_i, u_i)$ where $l_i$ and $u_i$ are the lower bound and upper bound in the jth generation for $j=1, 2, \ldots, t$; solution $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN_{sol}})$ is a compromise of the current $i^{th}$ solution $X_i^{t-1} = (x_{i1}^{t-1}, x_{i2}^{t-1}, \ldots, x_{iN_{sol}}^{t-1})$, $p_{Best}$ $P_i = (P_{i1}, P_{i2}, \ldots, P_{iN_{gen}}) \in \{X_i, X_i^{t-1}, \ldots, X_i^1\}$ which is a local best such that $F(P_i) \geq F(X_i^t)$, $g_{Best}$ $G = (g_1, g_2, \ldots, g_{N_{gen}}) \in \{P_i, P_{i2}, \ldots, P_{iN_{sol}}\}$ which is a global best such that $F(G) \geq F(P_i)$ for $i=1,2,\ldots,N_{sol}$.

$$x_i^t = \begin{cases} x_{i0} & \text{if } \rho \in [0, C_u) \\ p_y & \text{if } \rho \in [C_u, C_p) \\ g_j & \text{if } \rho \in [C_p, C_g) \\ x & \text{if } \rho \in [C_g, 1) \end{cases}$$

\hspace{1cm} (1)

PROCEDURE OF SSO

**STEP S0.** Generate $X_i$ randomly, let $t=1$, $P_i=X_i$, and $G=P_i$, where $F(P_i)=\text{MAX}_i \{F(P_i)\}$.

**STEP S1.** Let $i=1$.

**STEP S2.** Update $X_i$ based on Eq. (1) and calculate $F(X_i)$.

**STEP S3.** If $F(X_i)>F(P_i)$, let $P_i=X_i$; else, go to STEP S5.

**STEP S4.** If $F(P_i)>F(G)$, then let $G=P_i$.

**STEP S5.** If $i<N_{sol}$, then let $i=i+1$ and go to STEP S2.
STEP S6. If \( t = N_{\text{gen}} \), then \( G \) is the final solution and halt; otherwise, let \( t = t+1 \) and go to STEP S1.

3. FORMULATION OF THREE BENCHMARKS

The formulations of three benchmarks demonstrated in this study are introduced first in this section.

The First Constrained benchmark

This benchmark is introduced by Brakan and McCormick [23]. Following, several algorithms are applied to solve this benchmark [1-3, 5].

Two variables are identified and the variable vector is given by

\[ \mathbf{x} = (x_1, x_2) \]  

The formulation of the first constrained benchmark is displayed as follows:

**Min** \( f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 \)  

**Subject to** \( g_1(\mathbf{x}) = x_1 - 2x_2 + 1 = 0, \)  
\( g_2(\mathbf{x}) = -(x_1^2/4) - x_2^2 + 1 \geq 0, \)  
\[ -10 \leq x_i \leq 10, \quad i = 1, 2 \]

The Second Constrained benchmark

The various algorithms are applied to solve the second benchmark [1-2, 4, 6].

In this case, seven variables are identified: \( x_1, x_2, \ldots, x_7 \) and the variable vector is given by

\[ \mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \]  

The formulation of the second constrained benchmark is displayed as follows:

**Min** \( f(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \) \[ + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 \] \[ + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \]  

**Subject to** \( g_1(\mathbf{x}) = 127 - 2x_1^2 - 3x_2^4 - x_3 \) \[ - 3x_2^2 - 5x_4 \geq 0, \]  
\( g_2(\mathbf{x}) = 282 - 7x_1 - 3x_2 - 10x_3^2 \) \[ - x_4 + x_5 \geq 0, \]  
\( g_3(\mathbf{x}) = 196 - 23x_1 - x_2^2 - 6x_4^2 \) \[ + 8x_6 \geq 0, \]  
\( g_4(\mathbf{x}) = -4x_1^2 - x_2^2 + 3x_4x_5 - 2x_7^2 \) \[ - 5x_6 + 11x_7 \geq 0, \]  
\[ -10 \leq x_i \leq 10, \quad i = 1, 2 \]
The Third Constrained benchmark

Kannan and Kramer [24] propose the third constrained benchmark as shown in Fig. 1, a cylindrical vessel with two hemispherical heads is designed at both ends to minimize the fabrication cost. Several algorithms, such as MBA[1], CPSO[6], HPSO[7], GA1[8], GA2[9], NM-PSO[10], Q-PSO[11] and C-DE[12], are applied to solve this benchmark [1, 6-12].

Four variables are identified: \( T_s \) (\( x_1 \), thickness of the pressure vessel), \( T_h \) (\( x_2 \), thickness of the head), \( R \) (\( x_3 \), inner radius of the vessel), \( L \) (\( x_4 \), length of the vessel without heads):

\[
\mathbf{x} = (x_1, x_2, x_3, x_4) = (T_s, T_h, R, L)
\]

The variables \( T_s \) and \( T_h \) are considered as discrete variables multiples of 0.0625 in, and \( R \) and \( L \) are treated as continuous variables. The formulation is displayed as follows:

\[
\begin{align*}
\text{Min} & \quad f(x) = 0.6224x_1x_2 + 1.7781x_2^2 \\
& \quad \quad + 3.1661x_3^2 + 19.84x_4^2 \\
\text{Subject to} & \quad g_i(x) = -x_i + 0.0193x_3 \leq 0, \quad i = 1, 2, 3, 4 \\
& \quad g_5(x) = -x_1 + 0.0954x_4 \leq 0, \\
& \quad g_6(x) = -\pi x_1^2x_2 - (4/3)\pi x_3^2 \\
& \quad \quad + 1.296, 000 \leq 0, \\
& \quad g_7(x) = x_4 - 240 \leq 0, \\
& \quad 1 \leq x_i \leq 99, \quad i = 1, 2, 3, 4 \\
& \quad 1000 \leq x_4 \leq 2000, \quad i = 3, 4
\end{align*}
\]

The variables \( T_s \) and \( T_h \) are considered as discrete variables multiples of 0.0625 in, and \( R \) and \( L \) are treated as continuous variables. The formulation is displayed as follows:

\[
\begin{align*}
\text{Min} & \quad f(x) = 0.6224x_1x_2 + 1.7781x_2^2 \\
& \quad \quad + 3.1661x_3^2 + 19.84x_4^2 \\
\text{Subject to} & \quad g_i(x) = -x_i + 0.0193x_3 \leq 0, \quad i = 1, 2, 3, 4 \\
& \quad g_5(x) = -x_1 + 0.0954x_4 \leq 0, \\
& \quad g_6(x) = -\pi x_1^2x_2 - (4/3)\pi x_3^2 \\
& \quad \quad + 1.296, 000 \leq 0, \\
& \quad g_7(x) = x_4 - 240 \leq 0, \\
& \quad 1 \leq x_i \leq 99, \quad i = 1, 2, 3, 4 \\
& \quad 1000 \leq x_4 \leq 2000, \quad i = 3, 4
\end{align*}
\]

4. NUMERICAL EXPERIMENTS

The proposed SSO algorithm implemented in all experiments was coded in the C++ programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory. The experiments used 1000 generations (\( N_{gen} = 1000 \)), the number of solutions was \( N_{sol} = 100 \).

Experimental results of the first constrained engineering design benchmark are shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Benchmark 1: Comparison of the solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
</tr>
<tr>
<td>( x_2 )</td>
</tr>
<tr>
<td>( g_1(x) )</td>
</tr>
<tr>
<td>( g_2(x) )</td>
</tr>
</tbody>
</table>

Fig. 1 The pressure vessel design
Simplified swarm optimization for feasibility-based rules on constrained engineering design optimization

Yeh

Table 1: Feasibility-based results

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1(x) )</td>
<td>7.05E-09</td>
</tr>
</tbody>
</table>

In the second row of Table 1, the symbol * means the infeasible solution is obtained. The constraint \( g_1(x) \) = 7.05E-09 which is subjected to equal zero as shown in Eq. (4) falls into infeasible solution leads to the value of objective solution \(|f(x)|\) = 1.393454 obtained by the algorithm proposed by Bracken and McCormick [23] seriously loses. The constraint \( g_1(x) \) also falls into infeasible solution for the other four algorithms HS [2], GA [3], EP [5], and MBA [1]. Therefore, the optimum solution is \(|f(x)| = 1.39347741\) obtained by the proposed SSO algorithm with the best solution obtained at \( x = (0.822870567, 0.911435284) \), \( g_1(x) = 0 \), and \( g_2(x) = 6.7309E-06 \) which are all belonging to the feasible solutions.

Experimental results of the second constrained engineering benchmark are shown in Table 2.

Table 2: Benchmark 2: Comparison of the solutions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>2.330499</td>
<td>2.323456</td>
<td>2.326585</td>
<td>2.330499</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>1.951372</td>
<td>1.951242</td>
<td>1.950973</td>
<td>1.951372</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>-0.477541</td>
<td>-0.448467</td>
<td>-0.497446</td>
<td>-0.477541</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>4.365726</td>
<td>4.361919</td>
<td>4.367508</td>
<td>4.365726</td>
</tr>
<tr>
<td>( x_5 )</td>
<td>-0.624487</td>
<td>-0.630075</td>
<td>-0.618578</td>
<td>-0.624487</td>
</tr>
<tr>
<td>( x_6 )</td>
<td>1.038131</td>
<td>1.038660</td>
<td>1.043839</td>
<td>1.038131</td>
</tr>
<tr>
<td>( x_7 )</td>
<td>1.594227</td>
<td>1.605348</td>
<td>1.595928</td>
<td>1.594227</td>
</tr>
<tr>
<td>( g_1(x) )</td>
<td>4.46E-05</td>
<td>0.208928</td>
<td>1.17E-04</td>
<td>4.46E-05</td>
</tr>
<tr>
<td>( g_2(x) )</td>
<td>-252.561723*</td>
<td>-252.878859*</td>
<td>-252.400363*</td>
<td>-252.561723*</td>
</tr>
<tr>
<td>( g_3(x) )</td>
<td>-144.878190*</td>
<td>-145.123347*</td>
<td>-144.912069*</td>
<td>-144.878190*</td>
</tr>
<tr>
<td>( g_4(x) )</td>
<td>7.63E-06</td>
<td>-0.263414*</td>
<td>1.39E-04</td>
<td>7.63E-06</td>
</tr>
<tr>
<td>( f(x) )</td>
<td>680.63006</td>
<td>680.6413574</td>
<td>680.6322202</td>
<td>680.6300573</td>
</tr>
</tbody>
</table>

In the second row of Table 2, the symbol * means the infeasible solution is obtained. The constraints \( x_1 = -252.561723 \) and \( g_2(x) = -144.878190 \) which are subjected to be greater than or equal zero as shown in Eq. (10) and Eq. (11) fall into infeasible solution leads to the value of objective solution \(|f(x)| = 680.6300573\) obtained by the CA-DE algorithm proposed by Becerra and Coello [13] seriously loses. The constraints \( g_2(x) \) and \( g_3(x) \) also fall into infeasible solution for the other two algorithms IGA [4] and MBA [1]. And the constraints \( g_2(x) \), \( g_3(x) \) and \( g_4(x) \) fall into infeasible solution for the algorithm HS[2]. Therefore, the optimum solution is \(|f(x)| = 682.3060712\) obtained by the proposed SSO algorithm with the best solution obtained at \( x = (2.17932675, 1.879634988, -0.005798526, 4.547563084, -0.533158369, 1.133152247, 4.365727676) \), \( g_1(x) = 0.004486311 \), \( g_2(x) = 256.0247501 \), \( g_3(x) = 169.5636351 \), and \( g_4(x) = 32.11470036 \) which are all belonging to the feasible solutions.

Experimental results of the second constrained engineering benchmark are shown in Table 3.
In the second row of Table 3, the symbol \( ^* \) means the infeasible solution is obtained. The constraints \( x_1 = 0.7802 \) and \( x_2 = 0.3856 \) which are subjected to be greater than or equal 1 and less than or equal 99 as shown in Eq. (20) fall into infeasible solution leads to the value of objective solution \( f(x) = 5889.3216 \) obtained by the MBA algorithm proposed by Sadollah et al. [8] seriously loses. The constraints \( x_1 \) and \( x_2 \) also fall into infeasible solution for the other five algorithms GA1[8], GA2[9], CPSO[6], GQ-PSO[11] and C-DE[12]. The constraints \( x_1, x_2, \) and \( g_i(x) \) fall into infeasible solution for the algorithm HPSO[7]. And the constraints \( x_1, x_2, \) and \( g_i(x) \) fall into infeasible solution for the algorithm NM-PSO[10]. Therefore, the optimum solution is \( f(x) = 6008.410926 \) obtained by the proposed SSO algorithm with the best solution obtained at \( x = (84.99518193, 22.98158617, 172.6715902, 42.63405252) \), \( g_1(x) = -1.60185E-05 \), \( g_2(x) = -1.447539968 \), \( g_3(x) = -0.552476051 \), and \( g_4(x) = -10.74574313 \) which are all belonging to the feasible solutions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>0.8125(^*)</td>
<td>0.8125(^*)</td>
<td>0.8125(^*)</td>
<td>0.8125(^*)</td>
<td>0.8125(^*)</td>
<td>0.8125(^*)</td>
<td>0.7802(^*)</td>
<td>84.99518193</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>0.4375(^*)</td>
<td>0.4375(^*)</td>
<td>0.4375(^*)</td>
<td>0.4375(^*)</td>
<td>0.4375(^*)</td>
<td>0.4375(^*)</td>
<td>0.3856(^*)</td>
<td>22.98158617</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>40.3239</td>
<td>42.0974</td>
<td>42.0913</td>
<td>42.0984</td>
<td>41.6392</td>
<td>42.098411</td>
<td>42.0922</td>
<td>172.6715902</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>200.0000</td>
<td>176.6540</td>
<td>176.7465</td>
<td>176.6366</td>
<td>182.4120</td>
<td>176.6372</td>
<td>176.637690</td>
<td>198.4964</td>
</tr>
<tr>
<td>( g_1(x) )</td>
<td>-3.42E-02</td>
<td>-2.01E-03</td>
<td>-1.37E-06</td>
<td>-8.80E-07</td>
<td>3.65E-05</td>
<td>-8.79E-07</td>
<td>-6.67E-07</td>
<td>0</td>
</tr>
<tr>
<td>( g_2(x) )</td>
<td>-5.28E-02</td>
<td>-3.58E-02</td>
<td>-3.59E-04</td>
<td>-3.58E-02</td>
<td>3.79E-05</td>
<td>-3.58E-02</td>
<td>-3.58E-02</td>
<td>0</td>
</tr>
<tr>
<td>( g_3(x) )</td>
<td>-304.4020</td>
<td>-24.7593</td>
<td>-118.7687</td>
<td>3.1226(^*)</td>
<td>-1.5914</td>
<td>-0.2179</td>
<td>-3.705123</td>
<td>-86.3645</td>
</tr>
<tr>
<td>( g_4(x) )</td>
<td>-400.0000</td>
<td>-63.3460</td>
<td>-63.2535</td>
<td>-63.3634</td>
<td>-57.5879</td>
<td>-63.3626</td>
<td>-63.362310</td>
<td>-41.5035</td>
</tr>
<tr>
<td>( f(x) )</td>
<td>6288.7445</td>
<td>6059.9463</td>
<td>6061.0777</td>
<td>6059.7143</td>
<td>5930.3137</td>
<td>6059.7208</td>
<td>6059.7340</td>
<td>5889.3216</td>
</tr>
</tbody>
</table>

5. CONCLUSION

The SSO algorithm is applied to optimize three constrained engineering design benchmarks. The solutions of the constraints have been examined first as feasible solutions or infeasible solutions before the comparison of the objective solution obtained by each algorithm. Experimental results demonstrate the power of the proposed SSO for dealing with the feasible solutions of constraints. For all of the three constrained benchmarks, the computational results indicate that the proposed SSO obtains feasible solutions for all of the constraints while some infeasible solutions are obtained by the previously developed algorithms in literature [1-13, 23]. The value of the objective solution can be certificated only when the solutions of constraints are feasible solutions. Therefore, the proposed SSO provides optimized objective solutions than other algorithms in terms of examining the solutions of the constraints for feasibility or infeasibility. Moreover, the SSO algorithm can be applied to solve the constrained engineering optimization design especial for which strictly require feasible solutions of the constraints with the accuracy for the objective solution.

Total three famous constrained engineering optimization benchmarks are implemented in this work. In future studies, it will be considered to extend the scale and number of the benchmarks problem.
ACKNOWLEDGMENTS

This research was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C. under grant MOST 104-2221-E-007-061-MY3.

REFERENCES


[22] W.C. Yeh, Y. Z. Jiang, V. Y. Y. Chung, X. He, Solving reliability redundancy allocation problems with orthogonal simplified swarm optimization, International Joint Conference on Neural Networks (IJCNN 2015), Killaney, Ireland.